A Mobile Multi Agent System for Routing in Adhoc Network

M. A. Riahla[1], K. Tamine[2], A. Mifdal[2] and M. Mezghiche[1]
[1]*LIMOSE Laboratory, M'hamed Bougara University, South Campus 35000 Boumerdes, Boumerdes, Algeria*
[2]*XLIM Laboratory, Limoges University, Limoges, France*

Keywords: Dynamic Network, Mobile Agent, SMA, MANET, Routing Protocol, Distributed Algorithm.

Abstract: Dynamic networks are a challenge for the deployment of distributed applications on autonomous machines. But, these networks can meet problems with implementation of services such as routing and security in general. In this sense, the multi-agent systems are well suited for the design of distributed systems where several autonomous agents interact or work together to perform some set of tasks or satisfy some set of goals and moving the problem of analyzing from a global level to a local level and then reduce the complexity of the design (Ferber, 1997) In this paper we present a generic model Multi Agent system that we adapt to develop a new routing protocol for ad hoc networks. Wireless ad hoc networks are infrastructureless networks that comprise wireless mobile nodes able to communicate each other outside wireless transmission range. Due to frequent network topology changes in one hand and the limited underlying bandwidth in the other hand, routing becomes a challenging task. In this paper we present a novel routing algorithm devoted for mobile ad hoc networks. It entails both reactive and proactive components. More precisely, the algorithm is based on ant general behavior, but differs from the classic ant methods inspired from Ant-Colony-Optimization algorithm (Dorigo, Birattari and Stutzle, 2006). We do not use, during the reactive phase, a broadcasting technique that exponentially increases the routing overhead, but we introduce a new reactive route discovery technique that considerably reduces the communication overhead.

1 INTRODUCTION

Dynamic networks are a challenge for the deployment of distributed applications on autonomous machines. But, these networks can meet problems with implementation of services such as routing and security in general.

We have taken as a case study in ad hoc networks. For this we focused on Multi Agent who has a particular interest in the distributed problems in general and for which it is difficult to prevent all situations. Why the multi-agent systems are well suited for the design of distributed systems where several autonomous agents interact or work together to perform some set of tasks or satisfy some set of goals and moving the problem of analyzing from a global level to a local level and then reduce the complexity of the design. In this paper we present a generic model MultigAgent system that we adapt to develop a new routing protocol for ad hoc networks.

In multi-hop wireless ad hoc networks (MANETs) (Royer and Toh, 1999), mobile nodes cooperate with each other to form a network without a fixed infrastructure such as access point or base stations in which nodes perform routing discovery and routing maintenance in a self-organized way.

The routing is particularly a challenging task in MANETs. Indeed, because of the frequent changes in the network topology triggered by nodes displacements, establishment of new nodes connections and nodes disconnections, the routes discovery process is unstable. Practically speaking, efficient routes may quickly become inefficient or even unusable ones. To tackle this problem by ensuring a suitable routing through reliable algorithms, one important way is to update routing information more regularly than in wired networks.

However, this requires more routing control packets, which is specifically an issue in MANETs, since the bandwidth of the wireless medium is very limited and the medium is shared.

Beyond the routing overhead problem, our proposed protocol also attempts to solve the problems of packet delivery ratio and end-to-end delay. For this purpose, we propose a hybrid method

Riahla M., Tamine K., Mifdal A. and Mezghiche M..
A Mobile Multi Agent System for Routing in Adhoc Network.
DOI: 10.5220/0004697500310039
In *Proceedings of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2014), pages 31-39
ISBN: 978-989-758-000-0

that entails both proactive and reactive processes. The routes are established and periodically maintained with a constant number of mobile agents. The Agent is periodically created by each node and thus the number of agents in the network can continually be controlled. However, when a connection is planned to be established by a node with another one within the lack of a route in its routing table, the considered node makes a route request by setting a local variable available for Agents passing through it.

Our model is based on the ant behavior. A number of ant-based routing algorithms exist either in wired (Di Caro and Dorigo,1998) or in wireless (Caro, Ducatelle and Gambardella, 2005) (Correia, Vaza˜ and Lobo, 2009) (Correia and Vaza˜, 2008) (Laxmi, Jain and Gaur, 2006) (Bouazizi, 2002) networks. They are based on the pheromone trail laying-following behavior of real ants and the related framework of ant colony optimization (ACO (Dorigo and al, 2006). In all of these approaches, a source node broadcasts an Agent whenever it plans to build a route to a fixed destination. One of Agent roles is to deposit amounts of pheromone in order to mark optimal paths between a couple of nodes namely source and destination nodes. Unlike these methods, we do not exploit a broadcasting technique that exponentially increases the routing overhead, but we introduce a new idea through an ant-based algorithm that consists in setting a local route request whenever a node plans to send a data packet. It is the role of Agent, moving within the network during the proactive phase, to disseminate this information and to provide routes towards the requested destination. It should be noticed that our protocol doesn't deterministically establish the best route, since the Agent are not broadcast. However, the agents attempt to get as close as possible to the best route.

We begin by presenting some definitions on self-organized Multi Agent Systems and their analogy with dynamic networks. We present in the second part the principle of ad hoc networks and the problem of routing with this type of network data. In the Third part we present the architecture and mode of operation of a Multi Agent system adapted to generic case of dynamic systems. We detail the instantiations of the generic model: a routing protocol for ad networks. Before concluding, we discuss in Part VI results and tests of the new routing protocol using NS2 simulator.

2 MULTI-AGENT SYSTEMS AND DYNAMIC NETWORKS: ANALOGIES AND ISSUES

In recent years, a new topic of research has emerged: dynamic networks (also called autonomous systems). A dynamic and distributed network consists of a set of auto-configurable nodes that are constantly changing (the number of nodes and links change over time). The topology change is also one of the properties of these networks, because the network nodes can join and/or leave the network spontaneously. The main advantage of this type of network is the fast and inexpensive deployment and installation.

A multi agent system (MAS) is a set of agents operating in a common environment. This set of agents, not necessarily smart, is a complex system which appears a collective intelligence. This collective intelligence comes from the emergence of a global behavior of all agents. An example of this collective action is with the behavior of a colony of ants that act like an entities (ants) with no cognitive capacity but have achieved a high degree of organization and adaptation quote. An agent is a software entity (program) reactive, proactive and with social skills, able to act autonomously in its environment. Responsiveness refers to maintaining a constant link with the environment when a change occurs. The proactivity means that the system allows agents to generate and satisfy its goals. Social skills indicate that the system allows the agent to interact or cooperate with its environment and / or with other agents.

It is thus clear that there is an analogy between dynamic networks and multi-agent systems. Indeed, each node of a dynamic network is autonomous because it is not controllable by any other node on the network, is reactive because it can act as a server for other nodes, and can also be proactive in the case of his client node status, and finally it is social because it communicates and cooperates with other nodes in the network.

In this context, an inherent issue in the management of dynamic networks is: definition of a data routing protocol in the case of ad hoc networks.

3 ROUTING ISSUES IN AD HOC NETWORK

In multi-hop "wireless" ad hoc networks (MANETs) (Royer and Toh, 1999), "mobile nodes" cooperate with each other to form a network “without a fixed infrastructure” such as access point or base stations in which nodes perform routing discovery and routing maintenance in a self-organized way. Due to frequent network topology changes in one hand and the limited underlying bandwidth in the other hand, routing becomes a challenging task.

Several types of routing protocols have been specifically designed for ad hoc networks and have been classified into two main categories: reactive and proactive protocols. In reactive routing protocols such as AODV (Perkins and Royer, 1997) (Ad Hoc On demand Distance Vector) and DSR (Johnson Maltz and Broch, 2001) (Dynamic Source Routing), the routes are only discovered when required in order to save node and network resources, while in proactive routing protocols such as OLSR (Clausen and Jacquet, 2003) (Optimized Link State Routing Protocol) and DSDV (Perkins and Bhagwat, 1993) (Destination Sequenced Distance Vector) the routes are established in advance, avoiding consequently the delays that occur during the discovery of new routes. The problem raised by proactive protocols consists in the routing overhead, especially when there are frequent topology changes. This is highly inefficient when updating routes that rarely carry traffic. A reactive protocol is, in contrast, much more appropriate for such situations, since it generates lower overhead in terms of used bandwidth.

There is another kind of protocol that combines both reactive and proactive approaches called hybrid routing protocols. In this paper, we focus on a particular class of hybrid routing protocols based on an optimization technique known as ant colony optimization (ACO) (Caro, Ducatelle and Gambardella, 2005) (Correia, Vaz˜a and Lobo, 2009) (Correia and Vaz˜a, 2008) (Laxmi, Jain and Gaur, 2006) (Bouazizi, 2002) which are inspired from the foraging general behaviour of some ant species. The ant underlying behavior can be summarized as follows: ants deposit pheromone on the ground in order to mark some favorable paths that should be followed by other members of the colony, for instance, ants walking to and from a food source deposit on the ground a substance called pheromone. Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a significant effective way.

Several properties belonging to ant-based routing algorithms are strongly appropriate to address the problems inherent in MANETs: they are highly adaptive to network changes, robust to agent failures, and provide multipath routing. However, since they mainly rely on repeated path sampling, a significant overhead can be induced within native routing algorithms. Several ant-based routing algorithms for MANET have been proposed in state of the art previous work. However, within the attempt to limit the overhead caused by the ants, these algorithms considerably loose the inherent proactive sampling and exploratory properties belonging to the ants behavior adopted in the original ant-based algorithms.

4 A MULTI AGENT SYSTEM FOR GENERIC AND DYNAMIC NETWORKS

4.1 Architecture and Operating Mode for Multi Agent System

The Generic Multi Agent System that we propose to manage services in dynamic networks is composed of two types of agent communities (see figure 1). Both agent communities interact through different types of communication which we will present later.

The first community of agents is SMA_Node will manage the functions of mobile nodes in the dynamic network. Each agent will be called node, then represent a network node whose features will be explained below. Each node agent will be able to provide a set of network data packets are called packet. This set of data packets will be the second community of agents that will call SMA_Packet. Packet agents will travel along the dynamic network in a completely random way according to some metric contained in the node agents and packet agents at the interaction between these two types of agents.

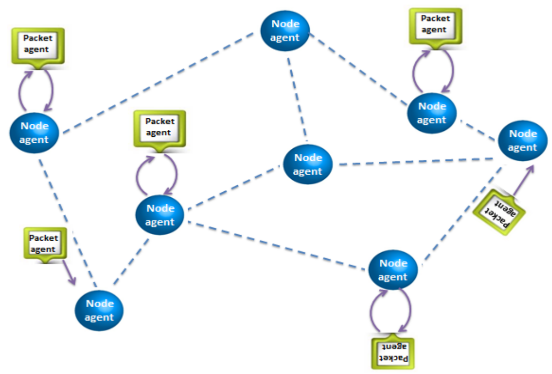

Figure 1: General scheme of our model.

Moving a packet agent to a node agent network is defined by a behaviour of packet agent which we will define later.

Each node agent SMA_Node from community will be defined generically by the following behaviours:

- Detect_neighbors () return Liste_node

This function allows the node agent to be able to see all the node agents that are its "scope" and with which it can communicate directly.

- Connect (v: node)

This function allows the node agent to start a connection with its neighbor node V in order to establish a communication with him.

- Disconnect (v: node)

This function allows the node agent to delete a connection with a neighbor node. This function also has the effect of removing the node agent v from its list of neighbor nodes.

- Connect ()

This function allows a node agent to join a network of node agents.

- Disconnect ()

This function disconnects the agent from network.

- Generate () return packet

This function allows the agent to create and distribute in the network a new packet agent.

- Read_info (p: packet, info: information)

This feature allows the agent to read and get information broadcast over the network and carried by the packet agent p as it moves from node to node.

- Write_info (p: packet, info: information)

This feature allows the agent to write information into the packet agent p so that it will be distributed in the network.

- Move (p: packet, n: node)

This function allows the node agent to send the packet agent p to another node neighbor agent (node agent n). It is this feature that allows the distribution of packet agents in the network.

Each agent from community SMA_packet will be defined generically by the following behavior:

- Create ()

This feature allows the packet agent to be created by the creator node agent;

- Delete ()

This feature allows the packet agent to be destroyed by its creator node agent;

- Transfert (n: node, info: information)

This feature enables the packet agent to transfer the information carried to the current node where it is located. This feature will be used by the current node agent to perform a read of the information in the network;

- Update_packet (n: node, info: information)

This function allows the packet agent to update its data with information provided by the current node agent. This feature will be used by the current node agent in order to broadcast information in the network;

- Choice_ displacement (n: node) return node

This feature lets you know randomly and according to some metric contained in the node agent n and in the current packet agent, the next node where the current packet agent will move;

- Move (init: node, final: node)

This function allows the packet agent to move from the node agent init to node agent final. The move action will be achieved through the move () function of init node agent.

4.2 Application of the Model in the Case of Routing in Ad Hoc Networks: PROTOCOL AgentRouting

The main idea of the protocol AgentRouting is to build a multi-agent based system where each node provides several kinds of agents. Regarding the purposes of the routing task, we design two main types of agents. A first mobile agent, called Ant-Agent, is responsible of establishing routes. A second mobile agent, called Rectifier-Ant, is issued by a node whenever a change in the network topology is detected. Our protocol is a complete multi-agent based system, where an agent works independently from the others. This fits very well spontaneous networks such as wireless ad hoc networks, because of the very high mobility and self-organization properties of this type of networks. Our protocol inherits from the advantages of this kind of model: autonomous work, distributed intelligence and robustness. Furthermore, the use of mobile agents allows to easily extending the functionalities of a protocol by simply adding other agents or by assigning other functionalities to existing ones.

AgentRouting protocol is based on a hybrid algorithm. In the proactive phase, the protocol uses mobile Agent as follows: each node (Origin-Node) periodically creates one Ant-Agent that moves across the network from one node to another and builds paths from the current node to its Origin-Node and paths from the current node to the last visited one when the Ant-Agent returns back to its

Origin-Node (the Ant-Agent has two phases: a Go-phase when it is sent by its Origin-Node and a Back-Phase when it returns to its Origin-Node).

When a data session is started between a source node s and a destination node d, s checks whether it has up-to-date routing information about d. If not, the node s makes a local route request. In our case, the route request is not broadcast to every node as it is the case in a classic ant routing protocol (Di Caro and Dorigo,1998) (Laxmi, Jain and Gaur, 2006) (Bouazizi, 2002) , but it is stored on the node s. The broadcast task is assigned to the Agent (created during the proactive phase and moving within the network) that have the responsibility to "intelligently" disseminate the route request throughout the network. These Agent gather information about the quality of paths they followed, and at their arrival in node s (which contains the route request to destination d), they return back to their source node by tracing back the path and by updating routing tables.

Before detailing our routing protocol, let's consider some of these assumptions. Each node must be able to broadcast hello messages to its one hop neighbors. We also consider that the links between the nodes of the networks are always bi-directional. Moreover, as the protocol must operate in an ad hoc environment where nodes are highly mobile, the routing protocol must take into account this constraint and be responsive to these frequent changes of the topology. Therefore, our routing protocol will consist of three modules: neighborhood discovery module, path discovery module and managing broken links module. In this paper, we focus our contribution on the two latter modules.

4.2.1 Proactive Phase

AgentRouting Protocol is a routing protocol for wireless ad hoc networks based on mobile Agent. To establish routes between nodes, our protocol uses mobile agents which are periodically created by each network node. An Ant-Agent belongs to a single node called Origin-Node. An Ant-Agent moves across the network from one node to another. When it reaches a node, the Agent establishes and builds, in its memory and in the routing table of the node, a path between this node and its origin node. Thereafter, the Agent chooses a next hop among its neighbors in a stochastic manner and proportionally to the amount of pheromone deposited by the other Agent during their Back-Phase.

To avoid routing loops, we assign a unique identifier <node_ID, Ant_ID> to agents, that is incremented at each creation of a new Ant-Agent. If a node receives several times the same Ant-Agent, it accepts the information given by the first one and ignores the others.

In order to monitor its *Ant-Agent*, a node assigns a configurable Time To Live (TTL) to agents whose value is proportional to the network dimension and is decremented at each hop. This means that an *Ant-Agent* will have two phases during its life cycle: G*o-Phase* where the agent builds a path from the current node to its O*rigin-Node* and *a Back-Phase* where the agent follows a reverse path from the one followed during its first phase (the *Ant-Agent* saves in its memory a reverse path during its *Go-Phase)*. At each node visited during the *Back-Phase*, the *Ant-Agent* builds and stores in the routing table of this node a path from this current node to the last node visited during the first phase (when TTL=0). This is the first step in the routing discovery process which is proactive.

4.2.2 Reactive Phase

During the proactive phase, a large number of paths are built. However, when a node s plans to send or forward data packets to an unknown destination node d, it triggers a route request process where the route request is locally saved. When an Ant-Agent, during its Go-Phase, visits a node which has made a route request (a node can have several route requests), the Ant-Agent switches to its Back-phase and deposits an amount of pheromone on each node of the reverse path towards its origin node. This mechanism is used to mark the paths towards the node s and thus inform the other nodes (and Agent) about this route request. The amount of pheromone deposited by the Ant-Agent is defined by the following equation (1):

$$Q_{it} = Q_{i(t-1)} + q \qquad (1)$$

Where Q_{it} is the pheromone level in the node n_i at time t and q is a positive constant (we choose q=0.1 for our simulations).

During its Go-Phase, a node chooses the next hop in a stochastic manner and proportionally to the amount of pheromones; this process increases the likelihood to select a path towards a claimant node without penalizing the other paths. Choosing the next node randomly and proportionally to the amount of pheromones allows us to increase the number of agents towards the claimant node s. On the one hand, this approach increases the chances of having an Ant-Agent issued from the destination node d (i.e., the Agent's source node is the destination d). On the other hand, it allows to quickly reaching nodes that have already established paths towards the destination d.

4.2.3 Stochastic Data Routing

In our protocol, the nodes stochastically forward the Agent. When a node has several neighbors concerned by nodes that made a route request, it randomly selects one of them with the probability p.

Each neighbour can have a quantity of pheromone related to nodes which made a route request.

Let's consider N (n) the set of n's neighbors and Q_{it} the amount of pheromone associated to a neighbor n_i stored in the routing table of the node n at time t.

The expression that gives the probability p to select a next hop n_j from node n is defined in equation (2).

$$P = Q_{it} / \Sigma Q_{kt} \quad (2)$$

In order to consider route requests in an equitable manner leading to a self-organizing system and a better management of frequent changes in the network topology, we propose to set up an evaporation process. This latter allows to no longer take into account the old route requests already satisfied. At each time interval, the amount of pheromone corresponding to each route request is decreased as defined in equation (3):

$$Q_{it} = (1-\alpha) * Q_{i(t-1)} \quad (3)$$

Where Q_{it} is the amount of pheromone related to a claimant node s, stored in the node n_i at time t and α is a real ($0<\alpha<1$) (we choose $\alpha= 0.1$ for our simulations).

5 TESTS AND SIMULATIONS RESULTS

5.1 Ad Hoc Simulations and Results

We evaluate our routing protocol through a serie of simulation tests. We compare its performance with AODV (Perkins and Royer, 1997), DSDV (Perkins and Bhagwat, 1993) and AntHocNet (Caro, Ducatelle and Gambardella, 2005).

We have evaluated our routing protocol under the NS2 environment.

In each simulation, which lasts 500s, the maximum speed in the scenario is fixed to 30m/s, the sending frequency of Agent is set to 0.5s, the evaporation frequency is set to 0.5s, the evaporation rate α is set to 0.1 and the updating value of pheromone q is set to 0.1. The traffic is randomly generated (the communications are established by randomly choosing pairs of nodes). A communication consists in sending 512 bytes packets by using UDP protocol. The total number of data packets ranges from 700 to 1000 packets per simulation.

The following experiments show a comparison of our protocol with three other routing baseline protocols: AODV (a reactive protocol), DSDV (a proactive protocol) and AntHocNet (an ant based routing protocol). For this purpose, we choose a network with an area of 500x500 m^2. The evaluation metrics used in the experiments are the following:

- The number of lost packets: this metric measures the number of packets which are not delivered to their destinations; it gives us, on the one hand, quantitative information about the robustness of our protocol and, on the other hand, information about the network congestion;
- The end-to-end delay: this metric represents the average delay between the packet sending time and its reception time;
- The total size of control messages generated by a protocol.

We have computed these metrics by varying, during the simulation time, the number of nodes, the number of data packets and the speed of nodes.

5.1.1 The Packet Loss

The packet loss metric allows us to evaluate and to compare the robustness and the effectiveness of the four protocols. Figure 2 and 3 show the variation of % packet loss considering respectively the variation of the number of nodes and the total number of data packets. It is clearly shown in these figures that our scheme, DSDV protocol and AntHocNet protocol have less packet loss than AODV protocol. The number of lost packets increases very quickly in networks of more than 70 nodes and in networks with high traffic. We can explain these results by the fact that the AODV protocol uses a broadcast mechanism that generates a very important overhead and this fact is aggravated by the proliferation of collisions, which overloads the system and generates a very large number of non-accomplished transmissions. In the case of our protocol, these results can be explained by the fact that we use an efficient reactive route discovery procedure instead of using a broadcast mechanism. Figure 4, which

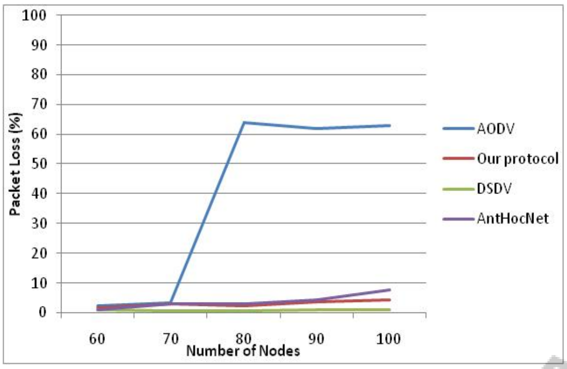

Figure 2: Packet loss ratio according to the number of nodes.

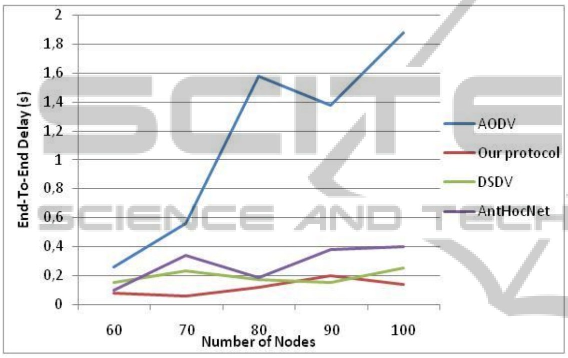

Figure 3: Packet loss ratio according to the total number of data packets.

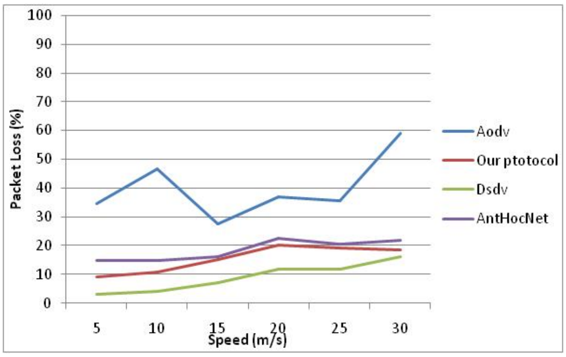

Figure 4: Packet loss ratio/nodes speed.

represents the variation of % packet loss considering the variation of nodes speed, shows that our protocol have less lost packets than AODV and AntHocNet protocols when we vary the nodes speed. This result may be partly explained by the fact that, in the case of our protocol, the link failures are partly supported by the proactive phase.

5.1.2 The End-to-End Delay

Figures 5, 6 and 7 show the average times of transmission in each protocol, considering respectively the number of nodes in the network, the total number of packets and the nodes speed. We can see that our protocol and the DSDV protocol generate less important delays than those generated by the AODV and AntHocNet protocols. Our protocol and DSDV protocol are both more efficient

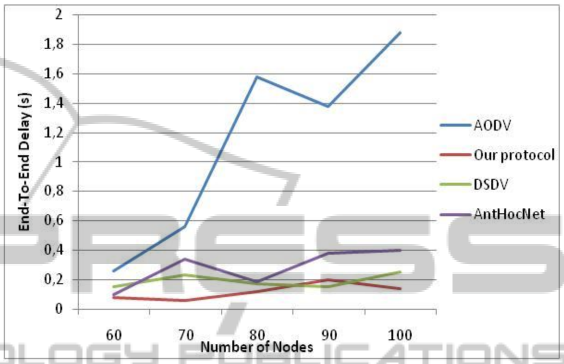

Figure 5: End-to-end delay (s) according to the number of nodes in the network.

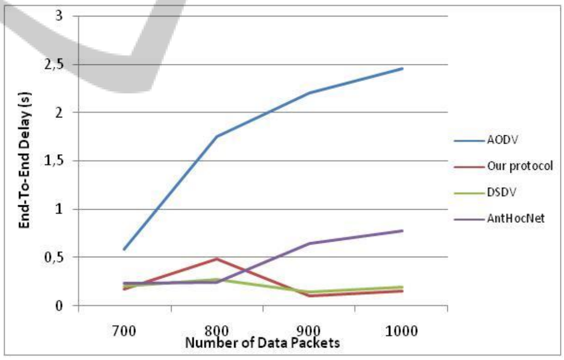

Figure 6: End-to-end delay (s) according to the total number of data packets.

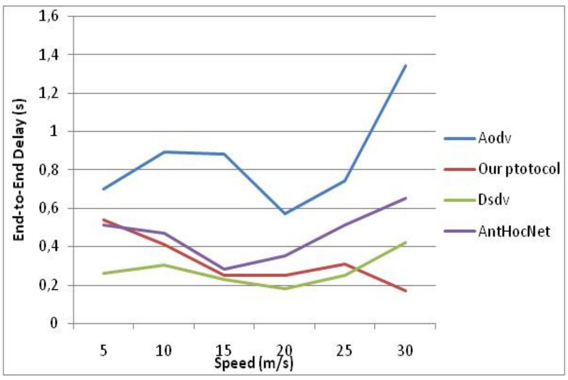

Figure 7: End-to-end delay (s)/speed (m/s) of nodes.

than AODV and AntHocNet protocols, especially in the case of a large number of nodes and data packets, and more specifically in the case of high nodes speeds. Regarding our protocol, this may be due to three main reasons:

- The hybrid character of our route discovery scheme reduces significantly the transmission times. Indeed, thanks to the proactive phase, it is no longer necessary to make, at each time, a route request; the route could exist in the routing table;
- As our protocol is multipath, it allows each node to have several paths (towards the same destination) whenever it wants to send a data packet;
- Unlike AODV protocol, our protocol does not require, during the reactive phase, a broadcasting technique; indeed, this latter exponentially increases the routing overhead and thus overloads the network; consequently it delays the paths establishment and the data packets delivery process.

5.1.3 The Communication Overhead

The packets that consume much bandwidth are those issued during the path discovery and maintenance phases. Therefore, we have measured the use of control messages by each studied protocol during these phases. Figures 8, 9 and 10 show the communication overhead in terms of, respectively, control packets size according to the number of nodes, the node activities and the speed of nodes.

We can see an important difference between our protocol and both AODV and AntHocNet protocols. The number of control packets increases slowly and linearly in our protocol, either by increasing the number of nodes or by increasing the network traffic, while it increases rapidly in the case of AODV protocol. This could be explained by the fact that in the case of our protocol, the number of agents is managed and controlled by each node and this number is still proportional to the number of nodes in the network and to the transmission frequency of agents. Besides, we avoid the broadcast technique that generates a lot of overhead. Instead, we use a more accurate and a more intelligent technique which only makes use of the available agents in the network. In AODV protocol, the number of control packets depends on many factors, including the number of route requests that consume much bandwidth, since they require a significant number of broadcasts. Moreover, we can expect an important number of collisions due to the broadcast technique

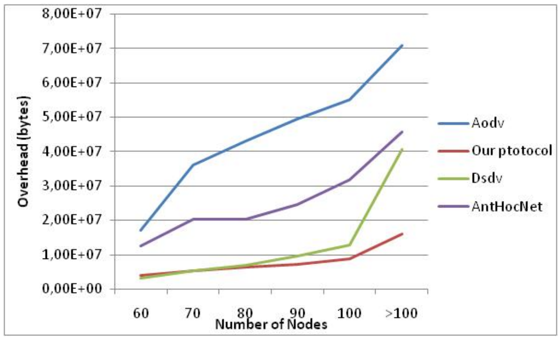

Figure 8: The communication overhead (total size (bytes) of control packets) according to the number of nodes in the network.

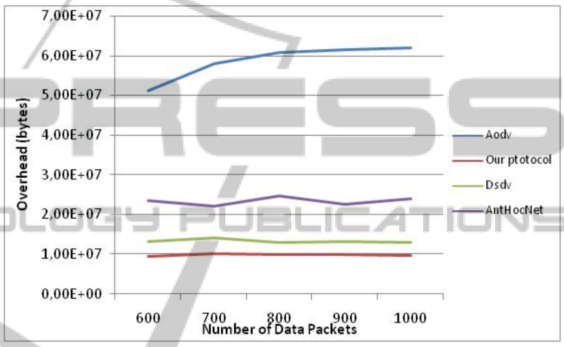

Figure 9: The communication overhead (total size (bytes) of control packets) according to the total number of data packets.

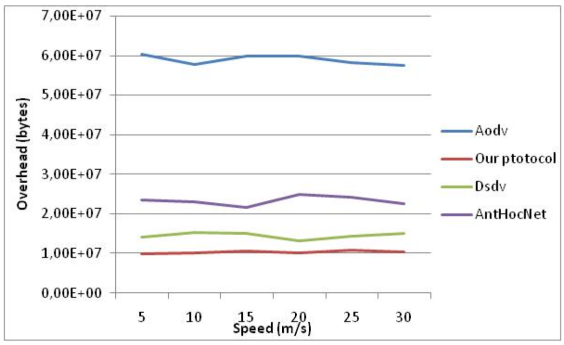

Figure 10: The communication overhead (total size (bytes) of control packets) /speed (m/s) of nodes.

and to the network density. The collisions lead to the increasing of the number of retransmissions, which consequently induces the increasing of the total number of packets in the network.

Figures 8, 9 and 10 show also that our protocol generates less overhead than DSDV protocol. This proves the effectiveness of our protocol, since the protocol DSDV is known to be efficient in this type of network.

6 CONCLUSIONS

In this paper we have presented a generic model of Multi-agent System dedicated to applications in dynamic networks. We have shown its feasibility in the case of the routing problem in ad hoc networks.
We have presented a novel ant-based routing protocol for MANETs. It is a hybrid algorithm combining reactive route setup with proactive route probing and exploration.

Considering the simulations results obtained using the proposed algorithm, we can argue that that our protocol reduces significantly the routing overhead and transmission delays. Regarding routing overhead, our algorithm out performs AODV and AntHocNet protocols which are supposed to generate fewer messages than the pure proactive routing protocols such as DSDV even if our protocol doesn't deterministically establish the best path. This latter fact is mainly due to the operating principle of our protocol: It doesn't require any broadcasting mechanism that leads to increase the number of control messages. Regarding latency, our algorithm allows also to achieve better results than AODV and AntHocNet. This proves the effectiveness of the proactive facet of our protocol and the usefulness of the underlying process that consists in computing several paths for the same destination.

In future work we plan to improve the exploratory working of quality of service. Our main idea relies on the assumption that the movement of Agent depends on the quantity of pheromone at each node which represents in our case study, the number of path requests in the network. Considering this assumption, we would design an intelligent distribution of Agent in the network according to the needs of each zone. More specifically, we plan to enhance our protocol with an interesting dynamic property leading to adapt itself according to the nodes activities in attempt to better exploit the Agent capabilities. The quantity of Pheromones could also be used in order to manage the number of Agent in the network and thus control the network congestion.

We also plan to use our multi-agent system to another dynamic environment as a P2P network; the principle is to use this model to ensure the anonymity of users in a P2P network.

REFERENCES

Bouazizi, I., (2002). The ant-colony based routing algorithm for manets. In *Proceedings of the 2002 international conference on parallel processing workshops, IEEE Computer Society*, Washington, DC, p. 79.

Caro, G. D., Ducatelle, F., & Gambardella, L. M., (2005). An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions ON Telecommunications*, 16, 443–455.

Clausen, T., & Jacquet, P., (2003). Optimized Link State Routing Protocol (OLSR), IETF RFC 3626.

Correia, F., & Vaza˜o, T., (2008). Simple ant routing algorithm Information Networking. *International Conference*, pp. 1–8.

Correia, F., Vaza˜o, T., & Lobo, V. J., (2009). Models for Pheromone Evaluation in Ant Systems for Mobile Ad-hoc Networks. In *Proceedings of the 2009 First International Conference on Emerging Network Intelligence (EMERGING '09), IEEE Computer Society*, Washington, DC, pp. 85-90.

Di Caro, G., & Dorigo, M., (1998). Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9, 317–365.

Dorigo, M., Birattari, M., & Stutzle, T., (2006). Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4), 28–39.

Ferber, J., 1997. Les systèmes Multi agents – Vers une intelligence collective.

Johnson, D. B., Maltz, D. A., & Broch, J., (2001). DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. In *Ad hoc networking*. Addison-Wesley Longman Publishing CO., INC., Boston, MA, pp. 139–172.

Laxmi, V., Jain, L., & Gaur, M. S., (2006). Ant colony optimisation based routing on NS-2. *International Conference on Wireless Communication and Sensor Networks (WCSN)*, INDIA.

Perkins, C., & Bhagwat, P., (1993). Highly dynamic destination sequenced distance vector routing (dsdv) for mobile computers pp. 234–244.

Perkins, C. E., & Royer, E. M., (1997). Ad hoc on-demand distance vector routing. In *Proceedings of the second IEEE workshop on mobile computing systems and applications*, pp. 90–100.

Royer, E. M., & Toh, C.-K., (1999). A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6, 46–55.